Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Linear Algebra and its Applications 403 (2005) 207–215

LINEAR ALGEBRA AND ITS APPLICATIONS

www.elsevier.com/locate/laa

A method for approximating symmetrically reciprocal matrices by transitive matrices

Geir Dahl *

Center of Mathematics for Applications, University of Oslo, P.O. Box 1053, Blindern, NO-0316 Oslo, Norway

Received 9 September 2004; accepted 2 February 2005
Available online 17 March 2005
Submitted by R.A. Brualdi

Abstract

The problem of approximating symmetrically reciprocal matrices by transitive matrices has received some attention recently. This problem has applications in multicriteria decision theory. Several approximation approaches have been suggested and analyzed. We here suggest another approach, called the *multiplicative approach*. We show that the optimal approximation in this sense may be found efficiently by transforming the problem into a known combinatorial optimization problem (the minimum cycle mean problem) for which efficient and simple combinatorial algorithms exist.

AMS classification: 15A48; 90C27; 91B06

Keywords: Transitive matrix; Symmetrically reciprocal matrix; Approximation

1. Introduction

A positive $n \times n$ matrix B is called *symmetrically reciprocal*, denoted by SR, if $b_{ii} = 1$ $(i \leqslant n)$ and $b_{ij} = 1/b_{ji}$ $(i, j \leqslant n)$. (A matrix is said to be *positive* if it is

* Tel.: +47 22 85 5835; fax: +47 22 85 4349.
E-mail address: geird@math.uio.no

0024-3795/$ - see front matter
doi:10.1016/j.laa.2005.02.002

entrywise positive.) SR matrices were introduced by Saaty [10] in connection with multicriteria decision making and he suggested an approach called the Analytic Hierarchy Process. In that setting B may represent pairwise comparisons of n different alternatives under a certain criterion. Thus, b_{ij} represents the *relative* importance (dominance, quality) of an alternative (decision) i over another alternative j for a fixed criterion. It is then natural to ask for numbers giving the absolute weight of each alternative that "explain" these relative comparisons best possible. This problem was discussed in [3,6] and in both papers a least squares approach was taken. We refer to [6] for a further problem background and references to related papers. The goal of this paper is to suggest another method for finding these absolute weights and we use a different distance measure than in the least squares approach. An introduction to the Analytic Hierarchy Process may be found in [12]. We mention that SR matrices also arise in input-output models in economics, see [7].

We follow [7] and say that a positive $n \times n$ matrix $A = [a_{ij}]$ is *transitive* provided that $a_{ik}a_{kj} = a_{ij}$ $(i, j, k \leqslant n)$. Every transitive matrix is also SR. (In fact, if $i = j = k$ we get $a_{ii} = 1$ and from the functional equation with $i = j$ we then get $a_{ik} = 1/a_{ki}$.) Several properties of transitive matrices and relations to the class of SR matrices were established in [7]. Every transitive matrix A has rank one and may be constructed as follows

$$A = A(y) := \begin{bmatrix} 1/y_1 \\ 1/y_2 \\ \vdots \\ 1/y_n \end{bmatrix} \begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix}, \tag{1}$$

where $y = (y_1, y_2, \ldots, y_n)$ is a vector with positive components. Thus, the matrix $A = [a_{ij}]$ is given by $a_{ij} = y_j/y_i$ $(i, j \leqslant n)$. We let $\mathscr{T}_n$ denote the set of all transitive $n \times n$ matrices.

An important problem is to approximate a given SR matrix $B = [b_{ij}]$ by a transitive matrix. A main application is the mentioned setting in multicriteria decision making. Then one asks for a transitive matrix, which represents consistent judgments of the pairwise comparison of the alternatives. Thus one wants to find positive numbers $y_1, y_2, \ldots, y_n$ such that $b_{ij} \approx y_j/y_i$ for each i, j. This problem was discussed in [6]. This approximation problem is not a straightforward task as the set of all transitive matrices is a complicated nonlinear manifold (in the space of real $n \times n$ matrices). However, by applying the entrywise logarithmic transformation to transitive matrices one obtains a linear subspace of the set of all $n \times n$ matrices since $a_{ik}a_{kj} = a_{ij}$ translates into $\log a_{ik} + \log a_{kj} = \log a_{ij}$ $(i, j, k \leqslant n)$. This transformation is exploited in [3] for the problem of minimizing $\|B - A\|_F$ where B is the given matrix and A runs through the set $\mathscr{T}_n$ of transitive matrices. In this paper we also use the entrywise logarithmic transformation, but to a different approximation problem. The structure of this problem makes it possible to take full advantage of the properties of the logarithmic transformation.

Remark. As kindly pointed out by L. Elsner there are several connections between this paper and a recent paper by Elsner and van den Driessche [5]. They also considered the problem of approximating an SR matrix S by a transitive matrix, but by minimizing the *relative error*

$$\max_{i,j} \left| \frac{s_{ij} - y_i/y_j}{s_{ij}} \right|,$$

over all $y_1, y_2, \ldots, y_n > 0$. This problem is then translated into a max-eigenvalue problem in a max-algebra: find a positive vector x and a number $\mu = \mu(S)$ such that $S \otimes x = \mu x$ where $S \otimes x$ is the vector whose ith component is $\max_k s_{ik}x_k$. It turns out that minimizing the relative error above is equivalent to minimizing the distance measure we propose in Section 2; this is a consequence of Theorem 2 in [5]. However, our analysis of the problem differs significantly from the one in [5]. We rely on a logarithmic transformation and convert the problems to a linear optimization problem with a certain combinatorial structure. Interestingly, the max-eigenvalue algorithm suggested in [4] is based on Karp's formula (5) (since $\mu(S)$ equals the maximum geometric cycle-mean of S, see [5]). As a result the final computational procedures of the approach in the present paper and in [5] are very similar.

Finally, in this introduction, we give some notation used throughout the paper. For matrices A and B of the same size we write $A \leqslant B$ (resp. $A < B$) if $a_{ij} \leqslant b_{ij}$ (resp. $a_{ij} < b_{ij}$) for all i, j. An all zeros matrix of suitable dimension is denoted by O.

2. The multiplicative approach

We first introduce a certain distance measure for positive matrices. Let A and B be two positive $n \times n$ matrices and define

$$\delta(A, B) = \inf\{\alpha \geqslant 1 : A \leqslant \alpha B, B \leqslant \alpha A\} \tag{2}$$

or, equivalently, $\delta(A, B) = \inf\{\alpha \geqslant 1 : (1/\alpha)a_{ij} \leqslant b_{ij} \leqslant \alpha a_{ij}\ (i, j \leqslant n)\}$. The infimum in (2) is attained; this and other basic properties of this distance measure are described next.

Lemma 2.1. *Let A and B be positive $n \times n$ matrices. Then the following statements hold.*

(i) $\delta(A, B) \geqslant 1$ *and* $\delta(A, B) = 1$ *if and only if* $A = B$.
(ii) $\delta(A, B) = \delta(B, A)$.
(iii) $\delta(A, B) = \max_{i,j} \max\{a_{ij}/b_{ij}, b_{ij}/a_{ij}\}$.
(iv) *If A and B are SR matrices*, *then*

$$\delta(A, B) = \inf\{\alpha \geqslant 1 : A \leqslant \alpha B\} = \max_{i,j} a_{ij}/b_{ij}.$$

Proof. (i) and (ii) follow directly from the definition (or from property (iii)). (iii) Let $\alpha = \delta(A, B)$. So $a_{ij} \leqslant \alpha b_{ij}$ and $b_{ij} \leqslant \alpha a_{ij}$ which gives $\alpha \geqslant \max\{a_{ij}/b_{ij}, b_{ij}/a_{ij}\}$. Since this holds for every i, j, we obtain $\delta(A, B) \geqslant \max_{i,j} \max\{a_{ij}/b_{ij}, b_{ij}/a_{ij}\}$, and equality must hold due to the minimality of α. (iv) Let A and B be SR matrices and assume that $A \leqslant \alpha B$ for some α, i.e., $a_{ij} \leqslant \alpha b_{ij}$ for each i, j. But then $1/a_{ij} \geqslant (1/\alpha)(1/b_{ij})$, and using the SR property we get $a_{ji} \geqslant (1/\alpha)b_{ji}$, so $b_{ji} \leqslant \alpha a_{ji}$. Therefore $B \leqslant \alpha A$ holds. This proves that the inequality $B \leqslant \alpha A$ is redundant in (2) for SR matrices and (iv) follows. $\square$

Consider again the problem of approximating a given positive matrix B by a transitive matrix, i.e., a matrix $A(y)$ as in (1). So we consider the following optimization problem:

$$\text{(TA)} \quad \phi(B) := \inf_{A \in \mathscr{T}_n} \delta(A, B) = \inf_{y > O} \delta(A(y), B), \tag{3}$$

where we ask for a transitive matrix $A(y)$ for which the distance $\delta(A(y), B)$ is smallest possible. We call this the *transitive approximation problem* and denote it by (TA). Note that in [6] one considered the nonlinear problem

$$\inf_{A \in \mathscr{T}_n} \|A - B\|_F,$$

where the Frobenius norm is used and a Newton method was suggested for solving it. Thus, these two approximation problem may look similar, but very different distance functions are used.

Returning to (TA) we see that $\phi(B) \geqslant 1$ and that $\phi(B) = 1$ if and only if B is a transitive matrix. We consider $\phi(B)$ as a measure on how well B can be approximated by a transitive matrix. We are mainly interested in this problem when the given matrix B is SR and hereafter we assume that this is the case.

Lemma 2.2. *Let B be an SR matrix. Then*

$$\phi(B) = \inf\left\{\alpha \geqslant 1 : y_j/y_i \leqslant \alpha b_{ij} \ (i, j \leqslant n), y_j \geqslant 1 \ (j \leqslant n)\right\}.$$

Proof. Since $A(y) = A(\lambda y)$ for each $\lambda > 0$, we may scale the variables in (TA) so that $y_i \geqslant 1$ for each i. Moreover, the approximation matrix $A(y)$ is transitive and therefore SR, so the result now follows from (iv) in Lemma 2.1. $\square$

In order to solve the optimization problem (TA) it is useful to make a transformation, the mentioned logarithmic transformation (we use natural logarithms). The inequality $y_j/y_i \leqslant \alpha b_{ij}$ then becomes $\log y_j - \log y_i \leqslant \log \alpha + \log b_{ij}$. By introducing the new variables $x_j = \log y_j$ $(j \leqslant n)$, $z = \log \alpha$ and the parameter $w_{ij} = \log b_{ij}$ the mentioned inequality becomes $x_j - x_i \leqslant w_{ij} + z$. Additionally, the inequalities $y_j \geqslant 1$ and $\alpha \geqslant 1$ mean that all the new variables $z, x_1, x_2, \ldots, x_n$ are nonnegative. Thus, the original problem (TA) is equivalent to the following optimization problem

$$\begin{array}{ll} \text{minimize} & z \\ \text{subject to} & x_j - x_i \leqslant w_{ij} + z \quad (i, j \leqslant n), \\ & z, x_j \geqslant 0 \quad (j \leqslant n). \end{array} \tag{4}$$

Let $\psi(B)$ denote the optimal value in this problem. So, if $(z, x_1, x_2, \ldots, x_n)$ is an optimal solution of (4) then $\alpha = e^z$, $y_j = e^{x_j}$ $(j \leqslant n)$ is optimal in the original problem (TA) and $\phi(B) = e^{\psi(B)}$.

The problem (4) is a linear programming (LP) problem, and therefore it is efficiently solvable, both theoretically and in practice. In the next section we show that the problem may be solved even more easily by exploiting an interesting relation to a known combinatorial optimization problem.

3. A relation to the minimum mean cycle problem

We describe how the linear programming problem (4) introduced in the previous section may be simplified.

First we introduce a combinatorial optimization problem of interest here. Consider a complete directed graph $G = (V, E)$ with vertex set $V = \{1, 2, \ldots, n\}$ and arc set $E = \{(i, j) : i, j \leqslant n, i \neq j\}$. Moreover, let $w : E \to \mathbb{R}$ be a given weight function where w_{ij} denotes the weight of arc (i, j). A directed cycle C in G is a sequence $(i_1, i_2), (i_2, i_3), \ldots, (i_p, i_1)$ of arcs where the only repeated vertex is i_1; we shall identify C with its set of arcs. The *weight* of a directed cycle C is $w(C) := \sum_{(i,j)\in C} w_{ij}$, and the *mean weight* is $\bar{w}(C) := w(C)/|C|$ where $|C|$ is the number of arcs in the cycle C. The *minimum mean cycle problem* is to find a cycle C with smallest possible mean weight $\bar{w}(C)$. This problem was introduced in [8] and plays an important role in some network flow algorithms. See [1] or [11] for a discussion of the problem. Karp [8] proved the following result: *the minimum mean weight of a cycle equals*

$$\min_{i\in V} \max_{0\leqslant k\leqslant n-1} \frac{d_n(i) - d_k(i)}{n - k}. \tag{5}$$

Here $d_k(i)$ is the minimum weight of a directed walk with exactly k arcs going from a vertex s of G to a vertex i using weight function w; the vertex s is a fixed, but arbitrary vertex of G. We can compute $d_k(j)$ using the Bellman–Ford shortest walk algorithm (see [1,11]) which is the following recursive relationship:

$$d_k(j) = \min_{i:(i,j)\in E} (d_{k-1}(i) + w_{ij}),$$

where $d_0(s) = 0$ and $d_0(j) = \infty$ for each vertex $j \neq s$. Since $d_k(i)$ can be found efficiently in this way, Karp's result immediately gives an efficient (polynomial-time) algorithm for solving the minimum mean cycle problem.

With this background we now return to the linear programming problem (4) introduced in the previous section. Let again G be the complete directed graph introduced above and define arc weights w_{ij} ($(i, j) \in E$) by

$$w_{ij} = \log b_{ij}, \tag{6}$$

where $B = [b_{ij}]$ is the given SR matrix that we want to approximate in (TA). Consider the following constraints from (4):

$$x_j - x_i \leqslant w_{ij} + z \quad ((i, j) \in E). \tag{7}$$

These inequalities are familiar from network flow theory, see e.g. [1] or [11, Chapter 8]. A vector $x = (x_1, x_2, \ldots, x_n)$ which satisfies (7) is called a *feasible potential* with respect to the "arc costs" $w_{ij} + z$. It is well known (see [11]) that a feasible potential exists for a given set of arc costs, say c_{ij} for each arc (i, j), if and only if the graph has no directed cycle with negative cost, i.e., a cycle C with $c(C) := \sum_{(i,j)\in C} c_{ij} < 0$. From this discussion it follows that

$$\begin{aligned}\psi(B) = {} & \text{the minimum value of } z \text{ such that } G \text{ equipped with arc costs } w_{ij} + z \\ & \text{has no negative cost directed cycle.}\end{aligned} \tag{8}$$

This expression gives a combinatorial interpretation of (the logarithm of) the approximation error $\phi(B)$ in the transitive approximation problem (TA). From this we may derive the following theorem concerning (TA).

Theorem 3.1. *Let B be a given SR matrix and let the associated weights w_{ij} be defined as in* (6). *Then $\psi(B) = \log \phi(B)$ is given by*

$$\psi(B) = -\min_C \bar{w}(C),$$

where the minimum is taken over all directed cycles C in G. This minimum may be found according to (5) *and therefore the* (TA) *problem may be solved efficiently using Karp's minimum mean cycle algorithm.*

Proof. Let the cost of the arc $(i, j) \in E$ be given by $w_{ij} + z$ and consider a directed cycle C in G. The cost of this cycle equals

$$w(C) + |C| \cdot z,$$

where $w(C) = \sum_{(i,j)\in C} w_{ij}$. This cost is nonnegative if and only if z satisfies

$$z \geqslant -w(C)/|C| = -\bar{w}(C).$$

Thus, G has no negative cost directed cycle if and only if

$$z \geqslant \max_C(-\bar{w}(C)) = -\min_C \bar{w}(C),$$

where the maximum and the minimum is taken over all directed cycles C in G. Combining this with (8) we see that the optimal value z in (TA) must be equal to $-\min_C \bar{w}(C)$. This proves the theorem. □

4. The algorithm and an example

Based on the previous sections, using the same notation, we may summarize our approach in the following way.

Algorithm 1

Input: an SR matrix $B = [b_{ij}]$ of order n.
Output: $y \in \mathbb{R}^n$ such that $A(y)$ is optimal in (TA).

1. Calculate weights $w_{ij} = \log b_{ij}$ $(i, j \leqslant n)$.
   Use the Bellman–Ford algorithm to calculate the minimum weight $d_k(i)$ of a directed walk with exactly k arcs going from vertex 1 to vertex i.
2. Find the minimum mean cycle weight $\min_C \bar{w}(C)$ using Karp's algorithm.
3. Define new arc weights $w'_{ij} = w_{ij} - \min_C \bar{w}(C)$.
   Use the Bellman–Ford algorithm to calculate the distance $d'(i)$ from vertex 1 to each vertex i using length function w'.
4. Let $x_i = d'(i)$ and $y_i = e^{x_i}$ $(i \leqslant n)$. Output $y = (y_i)$ and the matrix $A(y)$.

The correctness of the algorithm follows from Theorem 3.1 and the problem transformation discussed in Sections 2 and 3. We comment on the steps by relating them to the optimization problem (4). The optimal value $z = -\min_C \bar{w}(C)$ of (4) is found in step 2. It then remains to calculate the optimal values $x_1, x_2, \ldots, x_n$ and this is done in step 3. We here use the fact that the shortest path distances $x_i = d'(i)$ represent a feasible potential, i.e., satisfy the inequalities (7). In the final step we transform back from (4) to (TA) (inverting the logarithmic transformation).

Some further comments on this approach are summarized as follows:

- Algorithm 1 has complexity $O(n^3)$ and is fast in practice. We tested our approach by implementing the algorithm using MATLAB. As an illustration, for problems with $n = 100$ it takes about 0.25 seconds CPU time on a Dell 2650 computer (running Linux, 2.6 GHz processor). The algorithm is very simple to implement.
- The algorithm solves problem (TA), i.e., it finds a globally optimal solution. In some other approaches, see e.g. [6], one works with a more difficult nonconvex approximation problem and may numerically only find locally optimal solutions.

Finally we present a small example which is also found in [6]. The example is due to Saaty and concerns a multicriteria situation where one considers five criteria for national welfare (inflation, unemployment, growth, domestic stability, and foreign relations) and the corresponding SR matrix is

$$B = \begin{bmatrix} 1 & 3 & 5 & 4 & 6 \\ \frac{1}{3} & 1 & 4 & 4 & 6 \\ \frac{1}{5} & \frac{1}{4} & 1 & 2 & 2 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & 1 & 2 \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix}.$$

We here obtain $\phi(B) = 1.5651$ and the optimal solution $y = (1.0000, 1.9168, 4.8990, 6.2603, 9.3905)$ and the corresponding transitive matrix

$$A(y) = \begin{bmatrix} 1.0000 & 1.9168 & 4.8990 & 6.2603 & 9.3905 \\ 0.5217 & 1.0000 & 2.5558 & 3.2660 & 4.8990 \\ 0.2041 & 0.3913 & 1.0000 & 1.2779 & 1.9168 \\ 0.1597 & 0.3062 & 0.7825 & 1.0000 & 1.5000 \\ 0.1065 & 0.2041 & 0.5217 & 0.6667 & 1.0000 \end{bmatrix}.$$

It can be of interest to compare this result to the results in [6] (although this means comparing solutions for *different* approximation problems). As in [6] we present the reciprocals of the weights, and the weights are scaled so that $\sum_i 1/y_i = 1$. Consider the following table

	1	2	3	4	5
FLR	0.4027	0.3531	0.0895	0.0929	0.0617
Eig.	0.4767	0.2865	0.1029	0.0819	0.0520
Mult.	0.5020	0.2619	0.1025	0.0802	0.0535

The row denoted *FLR* contains $1/y_i^{FLR}$ $(i \leqslant 5)$ where y^{FLR} is the optimal weight vector obtained using the approach suggested by Farkas et al. [6]. The row denoted by *Eig.* contains $1/v_i^P$ $(i \leqslant 5)$ where v^P is the Perron eigenvector of B; this vector is often used in practice for these kind of problems (as suggested by Saaty). Finally, the last row, denoted by *Mult.*, contains $1/y_i$ were y is the optimal solution found in our multiplicative approach. Thus, at least in this example, the solution obtained from the multiplicative approach resembles that of Saaty's eigenvector method.

5. Concluding remarks

Although our derivation followed a very different path, as explained in Section 1, our multiplicative approach is closely related to the max-eigenvalue approach in [5]. In general the problem (TA) may have several optimal solutions, and the solution found by our algorithm may differ from the max-eigenvector. The fact that there may be several optimal solutions to the approximation problem was pointed out in the concluding remarks in [5]. We also refer to that same paper for a further comparison of the multiplicative approach/max-eigenvalue approach to Saaty's eigenvector method, for instance concerning sensitivity with respect to perturbations in the given SR matrix.

Our logarithmic transformation corresponds to replacing the max-algebra by the max-plus-algebra, and the latter subject is treated in the book [2] (including the relation between eigenvectors and graph theory). Moreover, Olsder et al. [9] discusses algorithms for finding eigenvectors in the max-plus-algebra via linear optimization.

Finally, we mention that our approach (and the one in [5]) has similarities to a method for data scaling which is presented in [1].

Acknowledgment

The author wants to thank Prof. Ludwig Elsner for several valuable comments on the strong connections to the max-eigenvalue problem and the approach in [5], as well as for suggesting several useful references.

References

[1] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[2] F.L. Baccelli, G.J. Olsder, J.P. Quadrat, G. Cohen, Synchronization and Linearity. An Algebra for Discrete Event Systems, in: Wiley Series on Probability and Mathematical Statistics, Wiley, Chichester, 1992.

[3] M. Chu, On the optimal consistent approximation to pairwise comparison matrices, Linear Algebra Appl. 272 (1998) 155–168.

[4] L. Elsner, P. van den Driessche, Modifying the power method in max algebra, Linear Algebra Appl. 332–334 (2001) 3–13.

[5] L. Elsner, P. van den Driessche, Max-algebra and pairwise comparison matrices, Linear Algebra Appl. 385 (2004) 47–62.

[6] A. Farkas, P. Lancaster, P. Rózsa, Consistency adjustments for pairwise comparison matrices, Numer. Linear Algebra Appl. 10 (2003) 689–700.

[7] A. Farkas, P. Rózsa, E. Stubnya, Transitive matrices and their applications, Linear Algebra Appl. 302–303 (1999) 423–433.

[8] R.M. Karp, A characterization of the minimum cyclemean in a digraph, Discrete Math. 23 (1978) 309–311.

[9] G.J. Olsder, K. Roos, R.J. van Egmond, An efficient algorithm for critical circuits and finite eigenvectors in the max-plus algebra, Linear Algebra Appl. 295 (1999) 231–240.

[10] T.L. Saaty, A scaling method for priorities in hierarchical structures, J. Math. Psych. 32 (1977) 234–281.

[11] A. Schrijver, Combinatorial Optimization—Polyhedra and Efficiency, Springer, Berlin Heidelberg, 2003.

[12] E. Triantaphyllou, S.H. Mann, Using the analytic hierarchy process for decision making in engineering applications: some challenges, Int. J. Indust. Eng.: Appl. Practice 2 (1) (1995) 35–44.